Corso di Abilità Informatiche A.A. 2024-2025

INTRODUZIONE ALL'INFORMATICA (Lezione 1)
Informatica: neologismo coniato anni '60 da Philippe Dreyfus: **elaborazione automatica dell'informazione**

Analisi dell'informazione:

- **Simbolica (o sintattica)**: analisi dei simboli e delle loro interazioni (sintassi) senza preoccuparsi del significato. Teoria dell'informazione di Claude Shannon. *Aspetti quantitativi* (es. concetto di entropia)
- **Semantica**: analisi del significato e del valore di verità dei simboli. Filosofia dell'informazione di Luciano Floridi. *Aspetti qualitativi* (es. «qual è la natura dell'informazione?»)

Elaborazione dell'informazione: cuore della disciplina **concetto astratto di calcolo:** una ***serie di azioni che consentono di manipolare un insieme di simboli***. Ha radici antichissime e nasce ben prima della matematica

Calcolo e scrittura > La scrittura cuneiforme, inventata nel Vicino Oriente alla fine del quarto millennio a.C., più antico sistema di scrittura conosciuto al mondo, deriverebbe da un antico dispositivo di conteggio
Si è riscontrato che nelle culture antiche esisteva: (1) Senso del numero; (2) Sistema di registrazione e rappresentazione dei numeri

Sistema additivo e posizionale

- **Sistemi di notazione additiva** (Egizi, Romani): un simbolo distinto per rappresentare ogni diversa unità. I, II, III, IV, V, VI, VII, VIII, IX, X …
- **Sistemi di notazione posizionale** (decimale): i valori rappresentati sono indicati dalla posizione del simbolo nella sequenza di caratteri che rappresentano il numero. 1, 2, 3,..., 10,..., 100,..., 1000

Abaco > Strumento in uso sin dal XXI sec. A.C. circa in Cina e Medio Oriente. Le prime evidenze di uso in Europa sono in Grecia intorno al V secolo A.C. Vi sono varie versioni (es. con o senza aste, basi numeriche differenti)

RADICI DELL'INFORMATICA: **Radici logiche** (filosofia, logica e matematica) e **Radici ingegneristiche** (fisica, meccanica)

RADICI LOGICHE
Gottfried Leibniz > Inventò il calcolo infinitesimale indipendentemente da Newton, e la sua notazione è quella in uso generale da allora. Il sogno di Leibniz era quello di ***creare un linguaggio che, per mezzo di calcoli simbolici, potesse stabilire quali dei suoi enunciati erano veri e quali relazioni logiche intercorrevano fra essi***.

Progetto > ***Caratteristica***: Sistema simbolico simile a quello dell'algebra per rappresentare le idee ben definite
Calculus ratiocinator: regole per manipolare i simboli
Caratteristica universale: Questa caratteristica consiste in una certa scrittura o lingua che rappresenta perfettamente le relazioni tra i nostri pensieri. I caratteri sarebbero diversissimi da tutto ciò che è stato immaginato finora; si è dimenticato, infatti, che i caratteri di questa nuova scrittura devono servire all'invenzione e al giudizio, come in algebra e in aritmetica.

George Boole > ***Logica Booleana****:* Usando metodi algebrici, Boole dimostra la validità dei sillogismi. Si occupa di valori binari, solitamente rappresentati come 1 (vero) e 0 (falso)
Esempio:

- X = «pianta» e Y = «vivente»
- XY = «tutte le piante che sono esseri viventi»

Alan Turing> Matematico inglese (1923-1954), padre della scienza informatica. 1936 introduce un **modello astratto di calcolo** (basato sull'osservazione), che ancora oggi è l'idea di base dei sistemi informatici contemporanei.

RADICI INGEGNERISTICHE **Pascalina**: Macchina calcolatrice in grado di sommare e sottrare.

Telaio Jacquard: Macchina per la tessitura in grado di produrre tessuti con un disegno specificato, programmabile con delle schede perforate. Ha rivoluzionato la tessitura in Francia e ispirato il progetto dei calcolatori.
Macchina Analitica: nel 1837 Charles Babbage descrive macchina analitica — il primo design di computer moderno. **Prima programmatrice della storia**.
Colossus: Londra, 1943-1944: Tommy Flowers realizza Colossus, il primo calcolatore elettronico, a valvole termoioniche.
ENIAC (Electronic Numerical Integrator and Computer): realizzato da Mauchly-Eckert nel 1943-45 Pennsylvania University. Primo calcolatore *general purpose*.

- 18000 valvole, 30 tonnellate
- 5000 operazioni aritmetiche al secondo
- 6 persone (donne!) dedicate alla sua programmazione (spostando interruttori e cavi).

Architettura Von Neumann
Nel 1946 **John Von Neumann** descrive il modello di architettura moderna di computer usato ancora oggi (architettura di Von Neumann):

- **Processore** (CPU): opera trasformazioni sui dati
- **Memoria**: insieme di celle numerate (indirizzate), ognuna contenente un dato di base. Si divide in memoria centrale e di massa.
- **Interfaccia**: si divide in input e output. Gestisce interazione con ambiente esterno (monitor, tastiera, ...)
- **Bus**: canale di comunicazione unico che gestisce lo scambio di informazioni tra i sotto-sistemi.

Transistor: Nel 1947 nasce il **Transistor** — il "mattone" dei computer moderni. Un transistor non è altro che un **interruttore controllato elettronicamente** (anziché meccanicamente). Permette l'implementazione delle **porte logiche** (implementazione della logica Booleana) alla base dei processori. Successivi miglioramenti tecnologici hanno portato ad una **miniaturizzazione** progressiva dei transistor. *Transistor più piccoli = più transistor in meno spazio = computer più potenti (veloci) a parità di spazio.*
Legge empirica di Moore: il numero di transistor in un circuito integrato di dimensioni fissate (es. un tipico processore) raddoppia ogni circa due anni.

Oltre Moore: I computer digitali tradizionali sfruttano la fisica classica ma cosa fare una volta raggiunto il limite di Moore?

- **Cloud computing**: deleghiamo i calcoli ad un server esterno.
- **Quantum computing**: sfruttiamo diverse leggi della fisica.
- Computer quantistico: i mattoni di base non sono i bit (0/1) ma i qbit. Un qbit può trovarsi in una combinazione di 0 e 1 (non solo in uno dei due stati).

SISTEMI COMPUTAZIONALI (Lezione 2)
Modello di calcolatore astratto: Alan Turing descrive il comportamento di un computer in termini astratti osservando un essere umano che esegue un calcolo basato su un insieme finito di regole. La macchina di Turing è un modello astratto di calcolo, introdotto nel 1936, ed è ancora l'idea di base alla base dei sistemi informatici contemporanei.

Dall'**osservazione empirica** Turing deriva una serie di vincoli per il suo modello astratto di calcolo:

- Limite fisso alla quantità di spazio su carta che un essere umano può considerare in un dato momento.
- Solo un numero finito di simboli può essere scritto e utilizzato in qualsiasi calcolo.
- In qualsiasi momento i simboli possono essere scritti o cancellati.
- Lo stato mentale attuale insieme all'ultimo simbolo scritto determina cosa fare dopo.

Una **macchina di Turing** è composta da:

- Un nastro illimitato diviso in caselle, ognuno delle quali può contenere esattamente un simbolo di un determinato alfabeto finito.
- Una testina per leggere e scrivere simboli da un dato alfabeto sulle caselle.
- Una unità di controllo (controller) che si trova esattamente in uno dei molti stati finiti in un dato momento.

Esempio di computazione: «Quando la macchina è nello stato **R** e sta esaminando il simbolo **a**, sostituirà **a** con **b**, si sposterà di un quadrato a **destra** e passerà allo stato **S**». Può essere espresso attraverso simboli (es. una quintuple): R a : b -> S. Espressioni analoghe: R a : b <- S / R a : b ★ S

La macchina di Turing "in azione"

Problema: Dato un numero naturale, verificare se è pari o dispari.

- Numero scritto sul nastro (input):
- Nessun limite alla quantità di nastro disponibile.
- Simbolo speciale (quadrato vuoto) per la terminazione □.
- Output: quadrati tutti vuoti tranne uno che sarà 0 se l'input era pari, 1 se era dispari. Stati:
- **Q** = stato iniziale (dal quadrato più a sinistra)
- **E, O** = stati risultanti dalla scansione dei numeri, E (pari), O (dispari)
- **F** = stato finale

Implicazioni: La macchina di Turing ha conseguenze molto importanti:

- Qualsiasi cosa calcolabile attraverso un processo algoritmico può essere calcolata da una macchina di Turing (definizione precisa di algoritmo)
- Se dimostriamo che un determinato compito non può essere svolto da una macchina di Turing, possiamo concludere che nessun processo algoritmico può svolgere tale compito (tesi Church-Turing)
- Tutte le macchine (anche quelle odierne) possono essere ricondotte al modello di Turing

Macchina di Turing Universale: Una macchina di Turing capace di simulare il comportamento di qualunque macchina di Turing. Dato un nastro con con:

- Codice numerico di una macchina di Turing *M* e un numero che indica l'ingresso di *M* separati da una casella vuota
- La macchina ricava le istruzioni (quintuple) dal codice numerico ed esegue sul secondo numero tutto ciò che le istruzioni prescrivono

---

Calcolatore moderno: Due componenti principali dei calcolatori moderni

1. **Hardware**: componenti materiali (memoria RAM, disco, monitor, etc.)
2. **Software**: applicazioni, sistema operativo

Calcolatore anni '50: tecnologia a valvole; bassa affidabilità; elevati consumi energetici; linguaggio macchina.
Calcolatore anni '60: tecnologia a transistor; affidabilità più elevata; consumi minori.
Calcolatore anni '70: tecnologia a circuiti integrati; linguaggi di programmazione di alto livello (es., C).
Calcolatore anni '80: personal computer; compattezza; costi ridotti; calore ridotto.
Calcolatore dagli anni '90 in poi: dimensioni ridotte; consumi molto bassi; connettività di rete elevata; interfacce user-friendly; apps

Linee di tendenza: Miniaturizzazione; Aumento della potenza di calcolo (maggior numero di cores, comunicazione più rapida tra CPU e memoria); Riduzione del consumo di energia; Aumento della capacità di memoria; Aumento della connettività di rete; Semplificazione dell'interfaccia utente.

Architettura Von Neumann

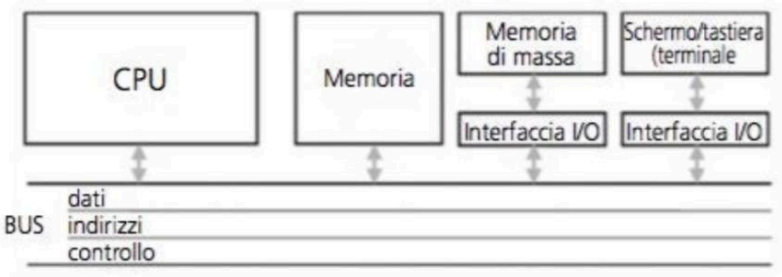
John von Neumann (1903-1957), è stato un matematico e informatico ungherese naturalizzato statunitense. Definisce un modello di **elaboratore elettronico** (architettura von Neumann), le cui caratteristiche sono:

- Modello di calcolatore universale
- *General purpose*: in grado di svolgere ogni tipo di calcolo
- Memoria condivisa tra dati e applicazioni.
- Codifica binaria dei dati e istruzioni.
- Separazione tra meccanismo di elaborazione e memorizzazione.

Componenti architettura Von Neumann

PROCESSORE (CPU)

- Opera ***trasformazioni sui dati*** (calcoli)
- ***ALU (Arithmetic Logic Unit)***: circuiti per operazioni aritmetiche-logiche elementari



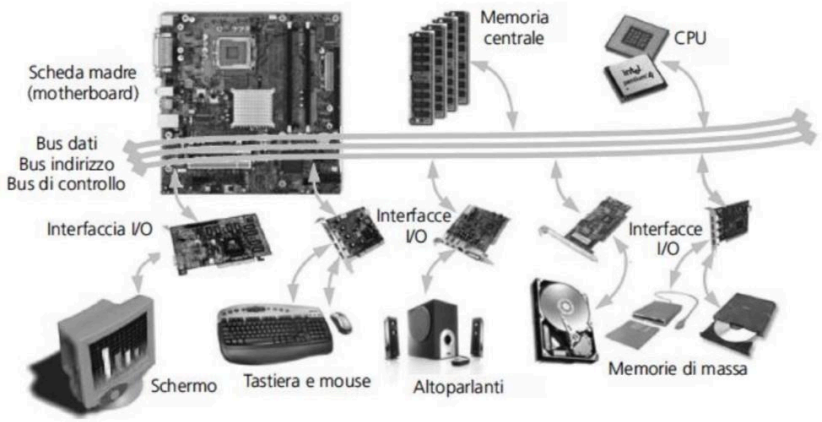
- ***Registri di memoria*** contenti: istruzione corrente; indirizzo di memoria istruzione
- ***Unità di controllo***: orchestra comunicazione tra ALU e registri/memoria

MEMORIA: insieme di celle numerate (indirizzate), con un dato di base contenuto in ogni cella.
Gerarchia della memoria:

1. **Memoria cache**: piccole dimensioni, velocissima, integrata nella CPU
2. **Memoria centrale RAM**: interagisce direttamente con la CPU, volatile
3. **Memoria di massa**: capiente ma più lenta (es. dischi magnetici)

INTERFACCIA: Si divide in ***input*** e ***output***. Gestisce interazione con ambiente esterno (monitor, tastiera, ...)

BUS: Canale di comunicazione unico che gestisce lo scambio di informazioni tra i sotto-sistemi.



Struttura del calcolatore
I personal computer (PC) tradizionali seguono abbastanza fedelmente l'architettura di Von Neumann.

- **Periferiche I/O**: tastiera, mouse, video, stampante, lettore DVD, etc.
- **Unità centrale** che contiene:
- ***Alimentatore***
- ***Ventole di raffreddamento***
- ***Scheda madre***. Slot ai quali vengono connessi il processore (CPU) e la memoria RAM

Classificazione dei calcolatori in base alla potenza di calcolo

1. **Embedded**: integrati dentro sistemi (es: semafori stradali), ottimizzati per eseguire un unico task.
2. **Minicomputer**: come i PC.
3. **Mainframe**: fascia alta, calcoli massivi. Usati da grandi organizzazioni
4. **Supercomputer**: tecnologie innovative, elevatissime prestazioni. Spesso usati per simulazioni in ambito scientifico

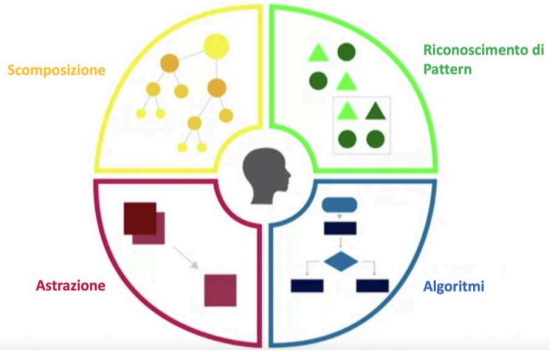
---

PENSIERO COMPUTAZIONALE PER LA RISOLUZIONE DEI PROBLEMI (Lezioni 3-4)
Il pensiero computazionale è la capacità di **immaginare e descrivere** un **procedimento costruttivo** che porti alla **soluzione di un problema**. Il concetto di pensiero computazionale fu introdotto per la prima volta da Seymour Papert nel libro Mindstorms, pubblicato nel 1980. Nel 2006, la prof.ssa Jeanette Wing ha reso popolare quest'idea sostenendo che il pensiero computazionale dovrebbe far parte delle **competenze di tutti i nuovi studenti** e definisce il pensiero computazionale come: ***il processo necessario per la formulazione e soluzione di problemi in forme comprensibili da agenti in grado di processare informazioni***.
Di che cosa parliamo? Il pensiero computazionale è una **capacità trasversale** che va sviluppata il prima possibile, non `e solo per informatici e programmatori. Serve a **trasformare un'intuizione in un procedimento costruttivo** che ci porta alla soluzione di un problema. Tale soluzione, inoltre, sarà talmente generica, da riuscire a riutilizzare quel ragionamento e quel procedimento ogni volta che ci troveremo ad affrontare un problema analogo (generalizzazione). Programmare è il modo migliore per acquisirlo.
Il pensiero computazionale è un approccio alla risoluzione dei problemi che *si basa su concetti e idee dell'informatica*; ci consente di affrontare un problema complesso, comprendere meglio in cosa consiste il

problema e sviluppare possibili soluzioni. Possiamo quindi presentare queste soluzioni in modo comprensibile a un computer, a un essere umano o a entrambi.
Non solo informatica > Il pensiero computazionale non è arido e schematico e non è legato soltanto e semplicemente all'informatica, in realtà è un concetto molto più ampio: è un'abilità (come scrivere, leggere e fare calcoli) che permette di acquisire elasticità mentale e capacità di risolvere problemi. L'attività di *coding* è la palestra in cui sviluppare e potenziare tale abilità.
Approccio > Il pensiero computazionale comporta:

1. la scomposizione di un problema in parti più piccole
2. la comprensione dei sottoproblemi
3. l'individuazione di quali informazioni sono necessarie per risolvere i sottoproblemi
4. lo sviluppo di una soluzione passo-passo

I quattro cardini del pensiero computazionale



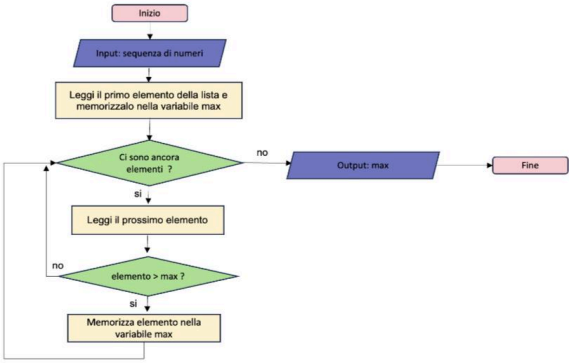
1- SCOMPOSIZIONE: Suddividere un problema o sistema complesso in parti più piccole, gestibili e semplici. Risolvere i sottoproblemi semplici e combinare le soluzioni per ottenere una soluzione del problema originale, complesso.
Obiettivo: Rendere il problema più affrontabile e lavorare su parti specifiche in modo sequenziale o parallelo.

2- RICONOSCIMENTO DI PATTERN: *Riconoscimento di modelli* significa trovare somiglianze o caratteristiche comuni all'interno del problema di partenza. La stessa soluzione può essere utilizzata per ogni occorrenza del pattern.
Esempio: Un matematico riconosce che l'equazione di una parabola ha una struttura comune, indipendentemente dai valori specifici dei coefficienti.
Obiettivo: Applicare soluzioni già sperimentate o adattare strategie conosciute per problemi simili.

3- RAPPRESENTAZIONE E ASTRAZIONE DEI DATI: Consiste nel determinare le caratteristiche del problema importanti e eliminare tutti gli aspetti non importanti. Usare tali caratteristiche per creare una rappresentazione astratta di ciò che stiamo cercando di risolvere.
Esempio: Per creare una mappa rappresentiamo le strade principali e i punti di riferimento più importanti, ignorando dettagli minori come i singoli alberi.
Obiettivo: Focalizzarsi solo sulle informazioni necessarie per risolvere il problema.

4- ALGORITMI: Un algoritmo è un insieme di istruzioni dettagliate su come risolvere un problema.
Identifica ciò che deve essere fatto (istruzioni non ambigue) e l'ordine in cui devono essere eseguite, in un numero finito di passaggi. Può essere descritto in italiano, mediante un diagramma di flusso o pseudocodice.
Gli algoritmi devono essere precisi, finiti, e risolvere un problema specifico in modo efficace. Sono la base per la creazione di programmi.
Cos'è un'istruzione? Un'istruzione può essere espressa in un linguaggio comprensibile ad un essere umano (detto *pseudocodice*) e poi viene tradotta in un linguaggio per il computer (detto *linguaggio di programmazione*).

Cosa può fare un computer? Un computer può eseguire solo operazioni binarie che richiedono due operandi, come aggiungere due numeri (1 + 1), moltiplicare due numeri, ($2 * 3$), confrontare due numeri ($2 < 8$).



Problema: Trovare il valore massimo in una sequenza di valori
Inizialmente, il valore massimo è il primo valore della lista. Quindi esaminiamo ogni valore nell'elenco. Se è maggiore del massimo attuale, quel valore diventa il massimo, altrimenti il massimo rimane lo stesso.
Dopo aver esaminato l'intero elenco, il valore massimo corrente è il massimo del valore nell'elenco.

Problema: Determinare se una sequenza di valori contiene un determinato valore
Analizzare ogni valore nella sequenza. Se è uguale al valore cercato allora abbiamo finito e possiamo concludere la ricerca. Se abbiamo analizzato tutti i valori della sequenza e non abbiamo trovato il valore cercato allora non è nell'elenco.

Pseudocodice
Descrizione informale, di alto livello, di ciò che fa l'algoritmo. È destinato agli esseri umani, piuttosto che alle macchine e non esiste una sintassi standard. Facilita la scrittura dei programmi.
Input: una sequenza di valori
Output: il valore massimo nella sequenza

Input: una sequenza di valori, e il valore da cercare nella sequenza
Output: vero se la sequenza contiene il valore cercato, falso se non lo contiene

PROGRAMMI
Un programma è un **insieme di istruzioni scritte in un linguaggio di programmazione** che specifica un'attività da svolgere da un computer. I programmi implementano algoritmi per risolvere problemi specifici.
Ruolo dei programmi: **traducono gli algoritmi in un linguaggio comprensibile per il computer**, consentendo l'esecuzione di compiti complessi.
Ruolo della programmazione: è l'arte di scrivere programmi per risolvere problemi.

Esempio: Trovare il valore massimo in una sequenza

Dall'algoritmo al programma

- **Input**: una sequenza di valori
- **Output**: il valore massimo nella sequenza

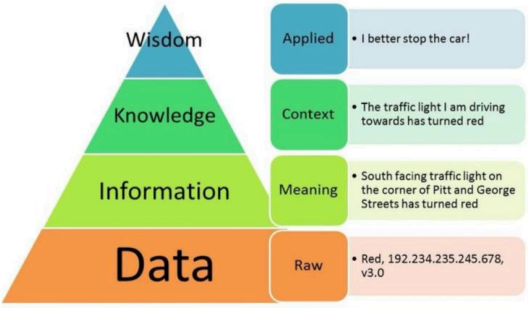
max ← primo elemento della sequenza
Per ogni *elemento* nella *sequenza*
se *elemento* > *max*
allora *max* ← *elemento*
restituisci *max*

```python
def find_max(sequence):
    # Inizializza il massimo al primo elemento della sequenza
    max_value = sequence[0]

    # Itera attraverso gli elementi della sequenza
    for num in sequence:
        # Se trovi un elemento maggiore del massimo attuale, aggiorna il massimo
        if num > max_value:
            max_value = num

    # Ritorna il valore massimo trovato
    return max_value

# Esempio di utilizzo del programma
sequence = [3, 7, 1, 9, 4, 2, 8]
print("Il massimo valore nella sequenza è:", find_max(sequence))
```

RAPPRESENTAZIONE E GESTIONE DEI DATI (Lezione 5)

La piramide della conoscenza

- **Dato** = valore (numerico) grezzo, senza un particolare significato associato. Esempio: 30
- **Informazione** = valore grezzo con una significato associato. Esempio: 'voto preso all'esame di storia moderna' oppure 'temperatura esterna'
- **Conoscenza** = insieme di informazioni collegate tra loro (sintesi di più informazioni). Esempio: se 30 è il voto di un esame, allora ho un esame in meno



NB. Non esiste una definizione assoluta per i concetti di dati, informazioni e conoscenza. Gli esperti possono non essere d'accordo o sottolineare aspetti diversi.

Rappresentazione dei dati
Senza perdita di generalità, un dato è un numero. Alcune rappresentazioni numeriche note:
- Sistema romano, non posizionale. X, IX, VIII,...
- Notazione posizionale decimale (cifre: 0,1,...,9)
- Notazione posizionale ottale (cifre: 0,1,...,7)
- Notazione posizionale esadecimale (cifre: 0,...,9,A,B,C,D,E,F)
- Notazione posizionale binaria (cifre: 0,1)

- Livello di mercurio sul termometro
- Numero di biglie in un sacchetto

Sistema binario > Il sistema binario è simile al sistema decimale che usiamo quotidianamente, ma invece di basarsi su 10 cifre (0-9), si basa su 2 cifre (0 e 1). Ogni posizione in un numero binario rappresenta una potenza del 2, partendo da 2^0 a destra. La ragione sta nel fatto che ***un dato di base è codificato tramite un segnale elettrico, che può essere presente (bit 1) o assente (bit 0)***. Il sistema binario è il più naturale in molti altri contesti. Esempio: scanalature CD/DVD (ci sono o non ci sono).
Possiamo contare solo da 0 ad 1, dopo di che abbiamo già finito le cifre a nostra disposizione, e siamo costretti ad inserire la seconda cifra per continuare a contare, ottenendo il numero 10, ma da lì possiamo solo contare fino ad 11 e poi abbiamo nuovamente finito le cifre, e dobbiamo ricorrere alla terza cifra, a formare il 100 e così via.

Decimale	Binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Decimale	Binario
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Codifica caratteri (file testo)
Codifica ASCII: *American Standard Code for Information Interchange* (Extended). È una codifica comunemente utilizzata per codificare i caratteri dei file di testo nel PC. Solitamente vengono usati 8 bit (1 byte) per simbolo (256 simboli / configurazioni possibili). Esempio:

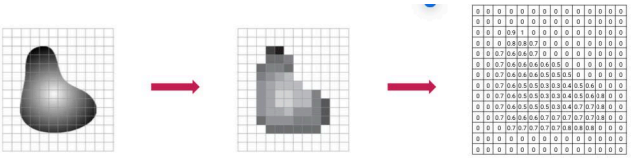
48 0x30 0011_0000 0	49 0x31 0011_0001 1	50 0x32 0011_0010 2	51 0x33 0011_0011 3	52 0x34 0011_0100 4	53 0x35 0011_0101 5	54 0x36 0011_0110 6	55 0x37 0011_0111 7	56 0x38 0011_1000 8	57 0x39 0011_1001 9	58 0x3a 0011_1010 :	59 0x3b 0011_1011 ;	60 0x3c 0011_1100 <	61 0x3d 0011_1101 =	62 0x3e 0011_1110 >	63 0x3f 0011_1111 ?
64 0x40 0100_0000 @	65 0x41 0100_0001 A	66 0x42 0100_0010 B	67 0x43 0100_0011 C	68 0x44 0100_0100 D	69 0x45 0100_0101 E	70 0x46 0100_0110 F	71 0x47 0100_0111 G	72 0x48 0100_1000 H	73 0x49 0100_1001 I	74 0x4a 0100_1010 J	75 0x4b 0100_1011 K	76 0x4c 0100_1100 L	77 0x4d 0100_1101 M	78 0x4e 0100_1110 N	79 0x4f 0100_1111 O
80 0x50 0101_0000 P	81 0x51 0101_0001 Q	82 0x52 0101_0010 R	83 0x53 0101_0011 S	84 0x54 0101_0100 T	85 0x55 0101_0101 U	86 0x56 0101_0110 V	87 0x57 0101_0111 W	88 0x58 0101_1000 X	89 0x59 0101_1001 Y	90 0x5a 0101_1010 Z	91 0x5b 0101_1011 [	92 0x5c 0101_1100 \	93 0x5d 0101_1101]	94 0x5e 0101_1110 ^	95 0x5f 0101_1111 _

RAPPRESENTAZIONE DI IMMAGINI
Un'immagine digitale è **composta da pixel disposti su una griglia quadrata regolare** (il pixel è generalmente quadrato ma può assumere anche diverse forme) È differente da immagine analogica (es. fotografia) che è composta da pigmenti colorati (on in bianco e nero) molto piccoli e spazialmente irregolare.
Il processo di digitalizzazione consiste di 2 operazioni: 1) **Campionamento**; 2) **Quantizzazione**

1- CAMPIONAMENTO: Solitamente **si approssima l'immagine usando una griglia (matrice) con celle della stessa dimensione**.
Ogni campione è chiamato pixel.
Più la griglia di campionamento è fitta e migliore sarà il risultato (e maggiore sarà il numero di pixel generati)

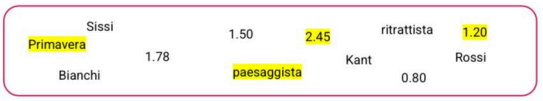
Multiplo	Sigla	Valore
exbibyte	EiB	2^{60} (Byte)
pebibyte	PiB	2^{50} (Byte)
tebibyte	TiB	2^{40} (Byte)
gibibyte	GiB	2^{30} (Byte)
mebibyte	MiB	2^{20} (Byte)
kibibyte	KiB	2^{10} (Byte)
byte	Byte	1Byte = 2^3 bit
bit	bit	

2- QUANTIZZAZIONE: È la **conversione dell'immagine campionata in valori numerici**. Ad esempio, in un'immagine in bianco e nero, il bianco diventa '0' e il nero '1'. Con la quantizzazione l'immagine può essere elaborata da un computer.



Scala delle dimensioni
Attenzione: queste sono potenze di 2. Non confondere kibibyte con kilobyte!
Il kilobyte vale 103 byte. 103 byte è circa 210 = 1,024 byte, ma non esattamente!

Dai dati all'informazione: In riferimento ai sistemi informatici potremmo dire che i **dati** sono delle **sequenze binarie su un certo supporto di memoria**. Affinché i dati apportino contenuto informativo, acquistino significato e generino conoscenza, abbiamo bisogno di una chiave di interpretazione. Abbiamo bisogno di **organizzare o strutturare i dati**.

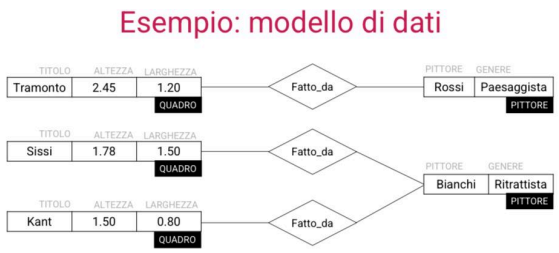


« Il quadro "**Primavera**" di "**Rossi**", un pittore del genere **paesaggista**, è alto **1.20** ed è largo **2.45**»

Modello di dati

Modello di dati ci consente di **attribuire una struttura ai dati** e di **rappresentare il dominio sul quale si sta lavorando**. Cos'è un modello?

- La ***rappresentazione di fatti relativi ad un fenomeno***, espressi in un qualche linguaggio formale.
- Il ***risultato di un processo di interpretazione*** mosso dalle idee e conoscenza dell'osservatore.
- ***Processo di astrazione***: si tralasciano gli elementi non importanti (approssimazione della realtà)



Vincoli di integrità : In generale i dati che si modellano sottostanno a dei vincoli di varia natura (vincoli di integrità). Ad esempio (vedi precedente):

1) Gli attributi 'Altezza' e 'Larghezza' devono essere numeri positivi
2) Non può esistere un quadro senza un pittore associato ad esso
3) Per ogni pittore ci dev'essere almeno un quadro

BASE DI DATI: **Insieme di dati strutturato secondo un certo schema**. I dati contenuti possono cambiare, aumentare o diminuire ma hanno sempre lo stesso schema (stessa struttura e soddisfanno sempre i vincoli di integrità). Caratteristiche:

- Deve essere **efficiente**, specialmente nel recuperare le informazioni.
- Deve essere **resistente**: un guasto non può portare alla perdita dei dati.
- Deve essere **integro**: mantenere i dati consistenti e corretti
- Deve essere **sicuro**: diversi utenti / diversi privilegi.

Tipi di basi di dati: A seconda del modello di dati usato, si distinguono diversi tipi di DB:

- **Relazionali**: Dati strutturati in tabelle.
- **Object-oriented**: Dati strutturati come collezione di oggetti, ognuno con attributi e relazioni con altri.
- **GIS – Geographic Information Systems**: Ottimizzati per immagazzinare dati spaziali/geografici e per recuperare informazioni in base alle relazioni spaziali tra gli oggetti.

Progettazione Base di dati

La progettazione di un database relazionale prevede quattro fasi principali:

1. **Analisi requisiti**: raccogliamo informazioni sul dominio (tramite interviste agli utenti).
2. **Progettazione concettuale**: costruiamo un modello del dominio, identificandone le entità e relazioni. Solitamente vengono usati i diagrammi ER (Entity-Relationship).
3. **Progettazione logica**: trasformiamo il diagramma ER in uno schema di tabelle (modello relazionale).
4. **Progettazione fisica**: implementiamo lo schema logico progettato al punto precedente con un linguaggio formale come SQL.

Progettazione concettuale: I modelli (o diagrammi) ER sono schemi concettuali grafici che ci permettono di modellare la conoscenza di un dominio al fine di crearne un database. Componenti:

- **Entità e loro classi**
- **Attributi associati alle entità e classi**
- **Relazioni tra entità**

Entità e proprietà

1. Entità: **oggetti di base che fanno parte del nostro modello** (es. pittori e quadri).
2. **Ogni entità ha una/più proprietà** (es. l'entità pittore potrebbe avere nome, data di nascita e n. quadri)
3. Ogni proprietà è contraddistinta da: **Nome (attributo)** e **Dominio**, l'insieme dei possibili valori che la proprietà può assumere.

Una **proprietà** può essere:

- **Atomica**: valore **non può essere scomposto in ulteriori sotto-valori**. (nome del pittore è atomico, mentre la data di nascita non lo è).
- **Unica**: ha **un solo possibile valore**. (Es: il n. di telefono di una persona non è necessariamente unico).
- **Totale**: Se **ogni entità ha un valore per questa proprietà** (Es: ogni persona ha un nome).
- **Parziale**: Se esistono **entità senza quella proprietà** (Es: non tutti hanno un numero di telefono)
- **Costante**: Se il suo **valore non cambia nel tempo** (Es: data di nascita di una persona).
- **Calcolata**: Se il suo valore **può essere calcolato a partire da altre proprietà** (Es: età può essere calcolato da data di nascita e data corrente).

Un'**istanza** di un'entità è un **valore specifico** dell'entità. Ad esempio: "Rossi" è un'istanza dell'entità "Pittore". "Tramonto" è un'istanza dell'entità "Quadri".

Una **chiave** è un insieme di **attributi di un'entità** tali che **due entità diverse hanno due diverse chiavi**.

- Esempio: {codice fiscale}, oppure {Nome, Cognome}

Una **chiave** è detta **primaria** se è quella che abbiamo deciso che verrà utilizzata nel database per identificare univocamente le entità. > In un diagramma ER, sottolineiamo la chiave primaria.

Tutte le **entità** del dominio che **condividono lo stesso tipo** sono **raggruppate in collezioni chiamate "classi"**.

- Esempio: le entità "pittori" e "musicisti" potrebbero far parte della classe "artisti".

Una **classe** è una **collezione di entità omogenee** (cioè tutte le entità aventi lo stesso tipo).
Le classi possono essere organizzate in modo gerarchico:

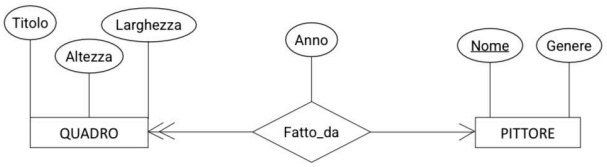
- Esempio: "pittori" può essere considerata una sotto-classe di "artisti".

Diagrammi Entità-Relazioni (RE)

I modelli (o diagrammi) ER sono **schemi concettuali grafici** che ci permettono di modellare la **conoscenza di un dominio** al fine di **crearne un database**.
Componenti:

- **Entità e loro classi**
- **Attributi associati alle entità e classi**
- **Relazioni tra entità**

Esempio:



Esempio modello relazionale:

Titolo	Autore	Altezza	Larghezza
Tramonto	Rossi	2.45	1.20
Tramonto	Verdi	1.50	0.80
Sissi	Bianchi	1.78	1.50
Kant	Bianchi	1.50	0.80

Tabella QUADRI

NomePittore	Genere
Rossi	paesaggista
Bianchi	ritrattista
Verdi	paesaggista

Tabella PITTORI

Nota: nel modello relazionale *relazione* è sinonimo di *tabella*.

Sistema basi di dati: Un modello di dati non fornisce solo strumenti per strutturare e vincolare i dati ma anche per manipolarli. Nelle basi di dati la manipolazione di dati consente di:

Esempio: implementazione schema logico
Esempi di implementazione attraverso l'uso di un linguaggio formale: SQL (Structured Query Language)

```
Create Table QUADRI
(Autore CHAR(49) not null,
Altezza DECIMAL (5,2),
Altezza DECIMAL (5,2))
```

```
Create Table PITTORI
(NomePittore CHAR(30) not null,
Genere CHAR(20) not null,
primary key (NomePittore))

insert into PITTORI
values ('Rossi', 'paesaggista')
```

```
select Titolo
from QUADRI

select
from QUADRI
Where Autore = 'Verdi'
```

PROTEZIONE DEI DATI (Lezione 6)
***DATI REAL-TIME*:**
Raccolta automatica e sistematica

- Infrastruttura web globale e pervasiva: apps, internet delle cose (IoT), dispositivi indossabili...
- I dati possono essere accumulati nel tempo

Accumulo a basso costo (rispetto alle pratiche di raccolta dati del passato)

- Procedure di misurazione automatizzate, meccanismi di incentivi
- Software di web scraping / Interfaccia di programmazione delle applicazioni (API) / Piattaforme di crowdsourcing

Dati pubblicamente disponibili

- La maggior parte dei dati sul web è pubblica / facilmente accessibile

DATI SOCIALI raccolti attraverso l'osservazione ('*observational data*')

- non derivanti da interventi progettati (esperimenti, sondaggi, raccolta dati amministrativi)
- rilevati da **interazioni sociali** "spontanee" e in modo automatico

Tracce di dati derivanti da **attività quotidiane** (post sui social media, ricerche web, acquisti online)

- opzioni di rinuncia non sempre facili (si pensi a eventi eccezionali come la pandemia)

DATI INTERCONNESSI
Interesse per l'insieme dei dati

- Emergenza di pattern o regolarità e creazione di modelli (predittivi)

Confronto tra oggetti

- Essere parte di un gruppo (o distribuzione) e discriminare tra gruppi (classificare)
- Somiglianze e differenze (ad es. identificare gli outliers)

Inferenza e analisi attraverso l'unione e la combinazione di dati, ad es.:

- I miei dati di viaggio possono rivelare più che i dettagli di viaggio (es. come la mia capacità d'acquisto)
- Dati sui social media per prevedere l'affidabilità creditizia (credit scoring)

Esempio: abitudini di acquisto
Una grande catena di vendita al dettaglio negli Stati Uniti ha utilizzato i dati d'acquisti dei clienti per prevedere le gravidanze delle clienti e inviare promozioni ben mirate e al momento giusto.
Sfida: "Se volessimo capire se una cliente è incinta, anche se non volesse farcelo sapere, è possibile farlo?"
Fatto: l'azienda ha saputo della gravidanza di una adolescente prima che lei lo comunicasse alla famiglia.

Uso e riuso dei dati

- I dati possono essere **condivisi e venduti** (ad es. *broker* di dati)
- **Scopi inaspettati** (anche abusi o inappropriati)
- Raccolta dati **"purpose-driven" vs. "possibility-driven"** (in caso di futuri usi possibili) > Ricerca scientifica / Marketing, analisi di mercato

Legge europea sulla protezione dei dati personali (GDPR): Il GDPR è entrato in vigore nel **2016** dopo essere stato approvato dal Parlamento Europeo. Impone obblighi alle organizzazioni, ovunque si trovino, che raccolgono dati relativi a persone nell'UE.

Dato personale (art. 4): qualsiasi informazione riguardante una **persona fisica identificata** o **identificabile** («interessato»); si considera identificabile la persona fisica che può essere identificata, direttamente o indirettamente, con particolare riferimento a un **identificativo** come il nome, un numero di identificazione, dati relativi all'ubicazione, un identificativo online o a uno o più elementi caratteristici della sua identità fisica, fisiologica, genetica, psichica, economica, culturale o sociale.

Trattamento (art. 4): qualsiasi operazione o insieme di operazioni, compiute con o senza l'ausilio di processi automatizzati e applicate a dati personali o insiemi di dati personali, come la **raccolta**, la **registrazione**, l'organizzazione, la **strutturazione**, la conservazione, l'adattamento o la **modifica**, l'**estrazione**, la consultazione, l'uso, la **comunicazione** mediante trasmissione, diffusione o qualsiasi altra forma di messa a disposizione, il **raffronto** o l'interconnessione, la limitazione, la cancellazione o la distruzione.

Principi generali

1. Liceità, correttezza e trasparenza
2. Limitazione delle finalità
3. Minimizzazione dei dati
4. Esattezza dei dati
5. Limitazione della conservazione
6. Sicurezza dei dati (integrità e riservatezza)
7. Responsabilità (accountability)

1- Liceità, correttezza e trasparenza: I **dati** devono essere raccolti in modo lecito, esempio: 1) **Consenso** dell'interessato; 2) Necessità di concludere un **contratto**; 3) Necessità di salvaguardare gli **interessi** del soggetto. Il **trattamento** dev'essere effettuato in modo corretto e trasparente: 1) Informazioni chiare sulle **finalità** e le operazioni del trattamento; 2) Informazioni sui **potenziali rischi** associati; 3) Diritto di **accedere** ai propri dati
Consenso informato (art. 4): qualsiasi manifestazione di volontà libera, specifica, informata e inequivocabile dell'interessato, con la quale lo stesso manifesta il proprio assenso, mediante dichiarazione o azione positiva inequivocabile, che i dati personali che lo riguardano siano oggetto di trattamento.
Condizioni per un consenso significativo:

- comprensione di cosa potrebbe essere fatto con i dati (linguaggio)
- comprensione di come i dati verranno utilizzati

Alcuni limiti del consenso

- Mancanza di competenze nella comprensione delle tecniche di trattamento dei dati
- Documenti lunghi e scritti in linguaggio giuridico non favoriscono il consenso attivo
- Consenso passivo vs. consenso attivo
- Imprevedibilità degli usi futuri (ricerca scientifica)

2- Limitazione delle finalità: Le finalità del trattamento dei dati deve essere definita prima che il trattamento inizi. Non può esserci ulteriore trattamento dei dati che non sia compatibile con la finalità iniziale, salvo eccezioni, es: ricerca scientifica. – *'In molti casi non è possibile individuare pienamente la finalità del trattamento dei dati personali a fini di ricerca scientifica al momento della raccolta dei dati. Pertanto, dovrebbe essere consentito agli interessati di prestare il proprio consenso a taluni settori della ricerca scientifica laddove vi sia rispetto delle norme deontologiche riconosciute per la ricerca scientifica."* (Considerando 33 del GDPR).

3- Minimizzazione dei dati: Il trattamento dei dati dev'essere limitato a quanto necessario per perseguire una finalità legittima. I dati personali dovrebbero essere trattati solo se le finalità del trattamento non è ragionevolmente conseguibile con altri mezzi. Il trattamento dei dati non deve interferire in modo sproporzionato con gli interessi, i diritti e le libertà in gioco (equilibrio interessi).

4- Esattezza dei dati: Esattezza in tutte le operazioni del trattamento: i dati inesatti devono essere cancellati o rettificati tempestivamente = Necessità di controllare i dati regolarmente aggiornandoli per garantirne l'esattezza.

5- Limitazione della conservazione: I dati devono essere cancellati o anonimizzati non appena questi non siano più necessari alle finalità per cui sono stati raccolti. Nota: i dati archiviati a fini di interesse pubblico, ricerca

scientifica o storica o usi statistici possono essere conservati per periodi più lunghi a condizione di adeguate garanzie. (Esempio: presenza di misure tecniche e organizzative in grado di garantire il rispetto del principio della minimizzazione dei dati).

6- Sicurezza dei dati: Sicurezza e riservatezza dei dati personali sono fondamentali per prevenire effetti negativi. Le misure di sicurezza possono essere di natura tecnica e/o organizzativa. – Esempi: **anonimizzazione** e **pseudonimizzazione** L'adeguatezza delle misure di sicurezza dev'essere determinata caso per caso e valutata periodicamente.
Anonimizzazione: rimuovere identificatori espliciti (es. nome) in modo che gli interessati non siano più identificabili.

Name	Sex	Date of Birth	Nationality	Disease
John Reds	M	22/3/1956	American	Diabetes
Susan Ramirez	F	02/12/1991	Mexican	Eczema

Name	Sex	Date of Birth	Nationality	Disease
******	M	22/3/1956	American	Diabetes
******	F	02/12/1991	Mexica	Eczema

Pseudonimizzazione: sostituire gli identificatori espliciti con una stringa "anonima" (es. alfanumerica).

Name	Sex	Date of Birth	Nationality	Disease
John Reds	M	22/3/1956	American	Diabetes
Susan Ramirez	F	02/12/1991	Mexica	Eczema

Name	Sex	Date of Birth	Nationality	Disease
23459	M	22/3/1956	American	Diabetes
86742	F	02/12/1991	Mexica	Eczema

Rischio di identificazione: l''anonimizzazione si è rivelata una sfida.
Vedi il caso di studio di Latanya Sweeney:

- Negli anni '90 un'agenzia governativa del Massachusetts decise di aiutare i ricercatori rilasciando dati sulle visite ospedaliere.
- Per mantenere anonimi i record, rimossero identificatori espliciti (ad es. nomi, indirizzi, ecc.) e lasciarono i codici postali, il sesso e le date di nascita.
- Sweeney acquistò i registri elettorali della città di Cambridge, Massachusetts, e combinò questi dati con il dataset anonimizzato, riuscendo a identificare alcuni soggetti.

Netflix prize competition

- Nel 2006 Netflix lancia una competizione per migliorare il proprio algoritmo di raccomandazione (''collaborative filtering'') rilasciando un dataset di recensioni anonimizzato (+100M)
- Due ricercatori dell'università del Texas (Arvind Narayanan e Vitaly Shmatikov) hanno incrociato il dataset con alcuni dati pubblici di IMDB (Internet Movie Database)
- I ricercatori hanno dimostrato che se un *attaccante* sapesse anche solo le date approssimative in cui un soggetto ha fatto 6 recensioni, potrebbe identificare quella persona il 99% delle volte.

7- Responsabilità : I titolari e responsabili del trattamento

- devono adottare in modo attivo e permanente misure finalizzate alla protezione dei dati;
- sono responsabili della conformità alla normativa;
- devono essere in grado di dimostrare in qualsiasi momento agli interessati, al pubblico in generale e alle autorità di controllo che essi operano in conformità delle disposizioni sulla protezione dei dati.

Esempio: stop temporaneo di ChatGPT
Il garante della privacy ha rilevato:

- Mancanza di informativa agli utenti sui dati raccolti da OpenAI
- Trattamento di dati inesatti
- Assenza di filtro per la verifica dell'età degli utenti – nei termini di servizio si diceche ChatGPT è rivolto a utenti >13 anni.

INTRODUZIONE ALL'INTELLIGENZA ARTIFICIALE (Lezione 7)

Proposta di ricerca del workshop di Dartmouth, 1956

"We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that ***every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it****. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves".*

Definire l'IA

Thinking Humanly	**Thinking Rationally**
"The exciting new effort to make computers think … machines with minds, in the full and literal sense" (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning…" (Hellman, 1978)	"The Study of mental faculties through the use of computational models" (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason and act" (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990) "The study of how to make computers do things at witch, at the moment, people are better" (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents" (Poole et al, 1998) "AI… is concerned with intelligent behaviour in artefacts (Nilsson, 1998)

Linee di sviluppo

Prospettiva filosofica	**Prospettiva tecnologica**
"Weak AI" : una macchina opportunamente programmata può agire *come se fosse* intelligente (simulare una mente)	"General AI" : creazione di programmi in grado far fronte a diversi compiti che generalmente richiedono intelligenza (Human Level AI)
"Strong AI": una macchina opportunamente programmata può essere davvero intelligente (essere una mente)	"Narrow AI" : creazione di programmi per l'esecuzione di compiti specifici spesso trasformate in applicazioni commerciali (riconoscimento vocale, traduzione testi...)

L'IA contemporanea

"Artificial intelligence (AI) refers to ***systems designed by humans*** *that, given a complex goal, act in the physical or digital world by perceiving their environment, interpreting the collected structured or unstructured data,* ***reasoning on the knowledge derived from this data*** *and* ***deciding the best action(s) to take (according to predefined parameters) to achieve the given goal****."* (High-Level Expert Group on Artificial Intelligence 2019)
Intelligent behavior*: "deciding the best actions to achieve a given goal, either in the physical or the digital world".* (Cristianini et al. 2021, The social Turn of AI, AI & Soc)

IA SIMBOLICA

Elementi di base

- Conoscenza: simboli o espressioni simboliche (interpretabili dall'uomo)
- Regole (definite dall'uomo) per manipolare i simboli ed estrarre conoscenza

Strategia risolutiva

- osservazione comportamenti umani nella risoluzione di giochi o enigmi
- ridurre il problema di partenza in sotto-problemi (insiemi di stati o configurazioni)
- sotto-programmi ('operatori') che eseguono le trasformazioni da uno stato all'altro
- analisi mezzi-fini per ridurre la differenza tra lo stato attuale del problema allo stato desiderato

- Esempio problema da risolvere: 3 cannibali e 3 missionari
- General Problem Solver (Newell and Simon, 1961)

Stati
CURRENT STATE:
LEFT-BANK = [3 MISSIONARIES, 3 CANNIBALS, 1 BOAT]
RIGHT-BANK = [EMPTY]

DESIRED STATE:
LEFT-BANK = [EMPTY]
RIGHT-BANK = [3 MISSIONARIES, 3 CANNIBALS, 1 BOAT]

Operatori
MOVE (#MISSIONARIES, # CANNIBALS, FROM-SIDE, TO-SIDE)

IA SUB-SIMBOLICA (CONNESSIONISTA)

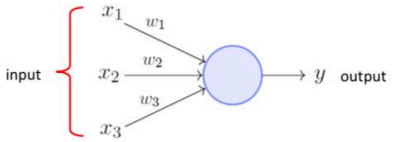
Elementi di base

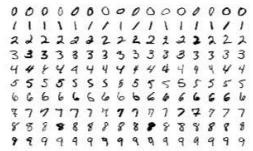
- unità connesse tra loro con input numerico e uno o più output
- Regole implicite, estratte dai dati attraverso una serie di equazioni

Strategia risolutiva

- prende ispirazione dal sistema nervoso e in particolare dal modello di neurone (McCulloch e Pitts, 1943)
- connessioni con diversa importanza ('pesi' diversi)
- un neurone somma gli input ricevuti dagli altri neuroni e si eccita se la somma (pesata) supera un determinato valore soglia

- Esempio problema da risolvere: riconoscimento di caratteri (o problemi di pattern recognition)
- Percettrone (Rosenblatt, 1957)





$$y = \begin{cases} 0 \ se \ w_1x_1 + w_2x_2 + w_3x_3 \leq S \\ 1 \ se \ w_1x_1 + w_2x_2 + w_3x_3 > S \end{cases}$$

- Obiettivo = calcolare il valore dei pesi (w_1 w_2 w_3)

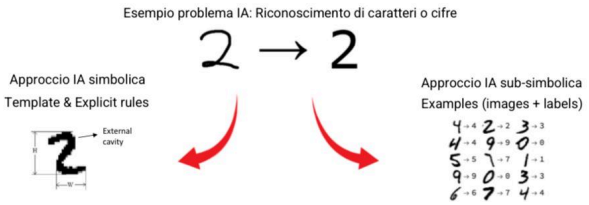
S = valore soglia

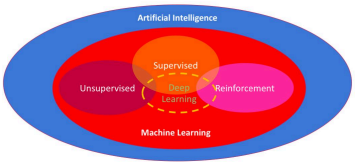
IA SIMBOLICA vs. IA SUB-SIMBOLICA

Aree di ricerca

Apprendimento automatico

Supervised learning
Unsupervised learning
Semi-supervised learning
Reinforcement Learning
Active Learning
Transfer learning
Adversarial machine learning
…


Esempio problema IA: Riconoscimento di caratteri o cifre
Approccio IA simbolica
Template & Explicit rules
External cavity
Approccio IA sub-simbolica
Examples (images + labels)



APPRENDIMENTO AUTOMATICO & RETI NEURALI (Lezioni 8 e 9)

APPRENDIMENTO AUTOMATICO

Una macchina è in grado di pensare? Definizione di intelligenza in termini di **fedeltà alla prestazione umana**: «Una macchina supererà il test se un esaminatore umano, dopo aver posto alcune domande in forma scritta, non sarà in grado di capire se le risposte provengono da una persona o no.»

Un problema... difficile! Per superare il test, un computer avrebbe bisogno delle seguenti capacità:

- **Rappresentazione della conoscenza** (Data Engineering): per memorizzare quel che conosce o sente
- **Ragionamento ed Apprendimento automatico**: utilizzare dati per individuare ed estrapolare schemi e trarre conclusioni
- **Visione artificiale**: per percepire il mondo
- **Interpretazione del linguaggio naturale** (NLP): per comunicare con l'uomo
- **Robotica**: per manipolare oggetti e spostarsi autonomamente

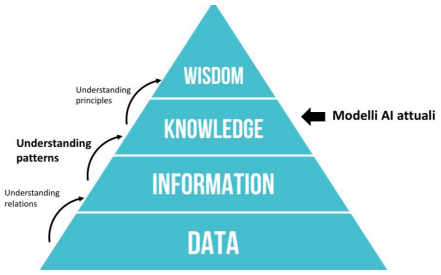
Apprendimento Automatico

1. ***Apprendimento***: Processo di acquisizione di nuove conoscenze, comportamenti, abilità.
2. ***Automatico***: Meccanismo o dispositivo capace di compiere determinate operazioni senza il diretto intervento umano.

Qual è stato il fattore trainante dello sviluppo dell'AI negli ultimi anni? Enorme avanzamento **nell'accesso ai dati**, reso possibile dai **big data**. Le 5 V dei big data descrivono i pilastri fondamentali che hanno sostenuto questa evoluzione. Ecco come si collegano allo sviluppo dell'AI:

1. **Volume**: L'**enorme quantità di dati generati quotidianamente** è uno dei motori principali per l'AI. Applicazioni come social media, dispositivi IoT, e-commerce e sensori industriali producono volumi di dati senza precedenti. Algoritmi di machine learning richiedono grandi quantità di dati per addestrarsi e migliorare la loro accuratezza.
2. **Varietà**: I **dati disponibili** oggi sono **molto diversificati**: immagini, video, testo, audio e dati strutturati. Questa varietà ha spinto lo sviluppo di modelli di AI più complessi, come le reti neurali profonde (Deep Learning), che possono analizzare dati multimodali e generare risultati avanzati (es. riconoscimento delle immagini o traduzione automatica).
3. **Velocità**: La rapidità con cui i dati vengono generati e processati ha migliorato le capacità dell'AI. Tecnologie come i sistemi di calcolo in tempo reale e l'elaborazione distribuita consentono agli algoritmi di analizzare i dati man mano che vengono prodotti, aumentando l'efficienza e permettendo applicazioni come il rilevamento delle frodi e i sistemi di guida autonoma.
4. **Veridicità**: La qualità e l'affidabilità dei dati sono fondamentali per il successo degli algoritmi di AI. I sistemi sono sempre più in grado di distinguere i dati utili da quelli rumorosi, grazie a tecniche avanzate di pre-elaborazione e a dataset curati che **assicurano la validità delle informazioni**.
5. **Valore**: L'AI è diventata centrale perché i **dati raccolti generano valore economico**. Gli insight derivati dai dati vengono utilizzati per ottimizzare i processi aziendali, personalizzare l'esperienza utente e prendere decisioni strategiche, dimostrando un chiaro ritorno sugli investimenti.

La rilevanza dei **big data** sta nella nostra capacità di **trasformarli in conoscenza**! Avere tanti dati non significa necessariamente aumentare la nostra conoscenza del mondo...Quando definiamo le **relazioni** che sussistono tra i dati attraverso la loro collezione e catalogazione otteniamo l'informazione. La conoscenza si ottiene scoprendo **correlazioni** nelle informazioni che abbiamo, congiuntamente all'interpretazione da parte degli individui.
Es. *Dato*: 37 / *Catalogazione*: temperatura corporea / *Correlazione*: se la temperatura supera i 37° hai la febbre.
Dopodiché si passa alla **saggezza** (***wisdom***): l'abilità di usare la propria conoscenza ed esperienza per fare buone decisioni e giudizi. Es. se hai la febbre e non ti senti bene, dovresti consultare un medico.



Un cambio di paradigma
La disponibilità di dati ha cambiato il modo in cui possiamo risolvere alcuni problemi.
- Come decidere se concedere un finanziamento ad un cliente?
- Data una foto, come determinare se è un cane o un gatto?

Diventare data-driven
L'informatica moderna ha visto un cambio di paradigma: dal concetto di algoritmo al concetto di dato. Grazie all'enorme disponibilità di dati, possiamo *risolvere nuovi problemi* e trovare connessioni nascoste attraverso l'apprendimento automatico.

Cosa possiamo fare con i dati?
- Le società di **carte di credito** monitorano ogni acquisto e identificano movimenti fraudolenti con un alto grado di precisione, utilizzando regole derivate dall'elaborazione di miliardi di transazioni passate.
- Gli operatori di **telefonia mobile** possono analizzare i modelli di chiamata degli abbonati per determinare, ad esempio, se la maggior parte dei loro contatti usa una rete rivale.
- Un dipartimento di **polizia urbana** ha sfruttato strumenti statistici per analizzare gli incidenti ed aiutare gli agenti a decidere quando e dove impiegare le risorse per prevenire la criminalità.
- Le aziende di **e-commerce** studiano le azioni dei clienti per suggerire offerte e suggerimenti su misura.
- Le **aziende sanitarie** generano informazioni mediche, organizzate in cartelle cliniche elettroniche.

Machine Learning: è la scienza del far agire i computer senza programmare esplicitamente.

- **Learning supervisionato (metodi predittivi)**: dati associati a delle **etichette**, utilizzate come esempi. *Task*: trova un modello che fa un buon lavoro nel prevedere l'output associato a un nuovo input (mai visto in precedenza).
- **Learning non supervisionato (metodi descrittivi)**: dati non contengono etichette. *Task*: trovare associazioni (o "pattern") interessanti nei dati: clustering,, regole di associazione, rilevamento di anomalie.
- **Reinforcement learning**: il modello sceglie automaticamente l'azione da intraprendere per massimizzare la ricompensa.
- **Semi-supervised learning**: alcuni dati associati ad un'etichetta e altri no. Obiettivo: associare l'etichetta ai dati «unknown».

Learning supervisionato: **(1) Training** (addestramento): il modello apprende i suoi parametri interni basandosi sulle etichette date. **(2) Testing**: verificare che il modello funzioni adeguatamente. Come?? **(3) Predizione**: dato un modello addestrato ed un nuovo input, il modello predice l'etichetta.

Qual è la differenza tra classificazione e regressione?

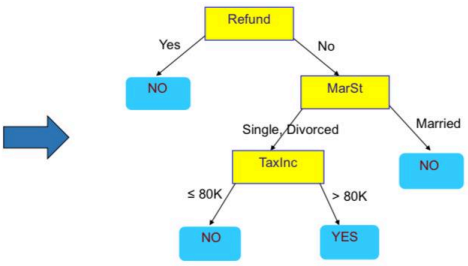
- **Classificazione**: le possibili etichette da associare ai dati sono un numero finito. Es: "cane", "gatto".
- **Regressione**: il valore delle etichette associate ai dati (quindi il valore da predire) è un valore numerico continuo. Es. la temperature misurata.

La compagnia Mega TelCo regalerà 100 minuti di chiamate ad una lista di clienti. Quanti minuti userà ogni cliente? È un problema di **regressione** perché per ogni cliente dobbiamo rispondere con un valore continuo da 0 a 100. Possiamo usare dati storici per addestrare il modello (regressore). Le etichette associate ai dati sono valori continui (numero di minuti).
*La compagnia Mega TelCo ha individuato una lista di possibili nuovi clienti. Chi accetterà una determinata offerta?*È un problema di **classificazione** perché per ogni nuovo cliente possiamo rispondere "sì" o "no"

Come sono fatti i dati input? Collezione di samples (clienti) con relativi attributi, un attributo è una proprietà o caratteristica di un oggetto e possono avere dei valori numerici o categorici. Ciascun sample ha una classe associata (yes o no in questo caso)
Si può adottare il modello ***decision tree / albero decisionale***: simile ad un diagramma di flusso; ogni nodo interno rappresenta un test su un attributo, ogni ramo il risultato del test, ogni foglia la classe finale predetta.
Le regole di classificazione sono determinate dai percorsi dalla radice alle foglie.

Tid	Refund	Marital Status	Taxable Income	Answer
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training set
(dati con etichetta di classe)

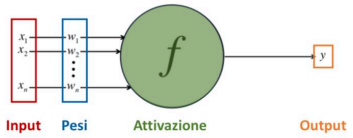


Un buon modello dovrebbe: (1) Catturare in modo accurato i «pattern» presenti training set. (2) Non essere troppo complesso! Idealmente vogliamo buoni risultati anche per casistiche che non ha potuto osservare in fase di apprendimento. **Un classificatore che «impara» perfettamente il training set è meno probabile che sappia generalizzare su nuovi dati.**

Come si valuta la qualità di un classificatore?

Training e Testing > 1ª regola per fare classificazione: **dividere il dataset in training set e test set**. Il training set viene usato per l'addestramento, mentre il test set viene "messo da parte" ed usato solo per la valutazione.
Il nostro obiettivo è costruire un modello che esamini i dati di addestramento e faccia **previsioni accurate su dati mai visti**. Quando un modello offre buone prestazioni su nuovi dati, è in grado di **generalizzare**. L'obiettivo è quello di costruire un modello che generalizzi il più possibile. Come possiamo controllare ciò? (1) Regolare la complessità del modello in modo che non impari a memoria i dati di training. (2) Avere dei dati di training ben distribuiti e rappresentativi.

RETI NEURALI: modelli di apprendimento automatico che si ispirano alla struttura del cervello. I mattoncini base sono i **neuroni**. Il neurone è l'elemento di base di ogni rete neurale. Ogni input X viene moltiplicato per il suo **peso w**; i risultati vengono **sommati** per produrre un unico valore; al risultato viene applicata la **funzione di attivazione** che produce l'**output y**.



$$y = f(x_1 w_1 + x_2 w_2 + \ldots + x_n w_n)$$

Come nel cervello umano, i neuroni sono interconnessi a formare una **rete organizzata a livelli**. I livelli non direttamente connessi all'input o all'output della rete sono chiamati *hidden* (nascosti) e sono molto importanti per il funzionamento della rete. Aggiungendo molti *hidden layers* otteniamo le reti profonde, chiamate **deep neural networks**

Universal Approximation Theorem: Le reti neurali sono in teoria **approssimatori universali**. Cosa significa? Data una **qualsiasi funzione** (anche molto complessa), **esiste** un modello di rete neurale che la approssima. Esempio: la funzione che prende in input i valori dei pixel di un'immagine e risponde «gatto» o «cane».

Come si addestra una rete neurale? Durante il training di una rete neurale, i valori dei pesi (w) di tutti i neuroni vengono modificati per **ridurre l'errore su un certo dataset.** Questo processo si chiama ***back-propagation***, e viene ripetuto molte volte durante il training.
Predizione: al termine del training abbiamo calcolato i pesi migliori che risolvono il nostro problema specifico.

INTRODUZIONE AL NATURAL LANGUAGE PROCESSING E AI GENERATIVA (Lezione 10 e 11)

Elaborazione del Linguaggio Naturale (NLP): NLP è un campo interdisciplinare tra informatica e intelligenza artificiale, che si occupa di fornire ai computer la capacità di elaborare dati codificati in linguaggio naturale.

- **Facilitare la comunicazione uomo-macchina**
  Esempi: risposte automatiche a domande (question answering), interazioni tramite chatbot, ecc.
- **Facilitare la comunicazione tra persone**
  Esempi: traduzione automatica, riassunti di testo, assistenza nella scrittura.
- **Analizzare e comprendere il linguaggio**
  Esempi: analisi sintattica, analisi del sentiment, classificazione dei testi.

Il NLP rappresenta quindi una tecnologia chiave per creare interazioni intelligenti e senza soluzione di continuità, facilitando la comunicazione e la comprensione del linguaggio umano.

L'elaborazione del linguaggio naturale ha le sue radici negli anni '40, soprattutto a causa della necessità di traduzioni automatiche emersa dopo 2GM. Fu con la pubblicazione di "Computing Machinery and Intelligence" di Alan Turing che si segnò un punto di svolta: propose il **test di Turing** come criterio per misurare l'intelligenza artificiale, introdusse quindi il compito dell'interpretazione e generazione automatica del linguaggio naturale.

Symbolic NLP (1950s-1990s): fino anni '80, la maggior parte dei sistemi di NLP si basava su complessi insiemi di regole **scritte a mano**. Il principio alla base del NLP simbolico può essere riassunto attraverso l'argomento della **stanza cinese** di John Searle. Dato un insieme di regole (ad esempio, un frasario cinese con domande e risposte corrispondenti), il computer simula la comprensione del linguaggio naturale applicando quelle regole ai dati che incontra. ***The Georgetown experiment***: esperimento di **traduzione automatica** in collab. con IBM; non era affatto un sistema completo: utilizzava solo **sei regole grammaticali** e **250 elementi lessicali** per tradurre **70 frasi** dall'inglese al russo.

Statistical NLP (1990s-2010s): rivoluzione data dall'introduzione di algoritmi di **machine learning** per l'elaborazione del linguaggio. Ciò fu reso possibile dal costante aumento della **potenza di calcolo** (legge di Moore), e dal graduale declino del predominio delle teorie linguistiche chomskiane che scoraggiavano l'approccio della linguistica dei corpora. Anni '90: primi successi significativi dei metodi statistici nell'NLP nel campo della **traduzione automatica**, grazie soprattutto al lavoro svolto presso IBM Research, come i modelli di allineamento di IBM. Questi sistemi riuscirono a sfruttare i **corpora testuali multilingue** esistenti, prodotti dal Parlamento del Canada e dall'UE, in seguito a leggi che richiedevano la traduzione di tutti i procedimenti governativi in tutte le lingue ufficiali dei rispettivi sistemi di governo. Un **modello di linguaggio** è un algoritmo che assegna probabilità a una sequenza di parole (unigram, bigram, trigram).

Neural NLP (2010s-oggi): nel 2003 il **percettrone multistrato** (addestrato su 14 milioni di parole) di Yoshua Bengio. Nel 2010 Tomáš Mikolov sviluppa Word2vec.

NLP Tasks: (1) **identificare/creare dati di addestramento** per il compito; (2) **selezionare uno o più benchmark** per la valutazione; (3) **identificare i modelli all'avanguardia** per il compito; (4) **calcolare le metriche** sui benchmark; (5) **confrontare i risultati**.

***The NLP pipeline*:** l'elaborazione del linguaggio naturale (NLP) si occupa principalmente di fornire ai computer la capacità di elaborare dati codificati in linguaggio naturale.
(1) Morfologia: Studia la struttura e la formazione delle parole. Analizza come i morfemi, le unità minime di significato, si combinano per formare parole, includendo processi come flessione, derivazione e composizione.
Lemmatizzazione: rimuovere desinenze inflessionali e restituire la forma base/del dizionario. **Analisi morfologica**: separare le parole nei morfemi individuali e identificare la classe dei morfemi.
(2) Sintassi: Si occupa delle regole e delle strutture che governano la combinazione delle parole per formare frasi grammaticalmente corretti. Studia come gli elementi si relazionano tra loro all'interno di una frase.
Parsing: determina l'albero di analisi grammaticale (*parse tree)* di una data frase.
(3) Semantica: È lo studio del significato delle parole, delle frasi e dei testi. Si concentra sull'interpretazione del contenuto linguistico e su come i significati sono rappresentati e relazionati nel linguaggio.
(4) Pragmatica (applicazioni NLP a livello superiore): Analizza come il contesto influisce sull'interpretazione del linguaggio. Esamina l'uso del linguaggio in situazioni concrete, includendo implicature, atti linguistici e il significato oltre il contenuto letterale.

Modello del linguaggio: è uno strumento che assegna delle probabilità a delle sequenze di parole

---

AI GENERATIVA & IMMAGINI (Lezione 12)
AI Discriminativa vs Generativa

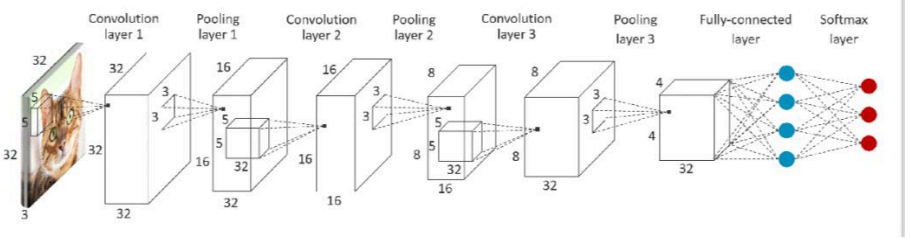
- **Discriminativa**: apprendono come discriminare le classi fornite nel dataset, ottimizzando un *decision boundary* (Text classification; object detection)
- **Generativa**: apprendono come generare nuovi dati con una distribuzione simile a quella vista nel training (Image generation; text-to-image; impainting)

***AI generativa* >** Utilizzo di **modelli di AI per produrre** diversi tipi di contenuto: testo, immagini, audio, video… Solitamente generano un output in base ad un certo input (prompt). **Apprendono in modo automatico pattern e struttura statistica dei dati di training, creando il nuovo contenuto "estraendo" dalla stessa distribuzione**. Di difficile realizzazione durante lo sviluppo "iniziale" del deep learning.

Harold Cohen, artista creatore di **AARON**, uno dei primi sistemi informatici progettati per creare arte sviluppato in diverse versioni dal 1972 al 2010. Non impara nuovi stili in autonomia: funziona definendo manualmente un set di regole nel programma. Può virtualmente produrre infiniti output nello stile predefinito. Risultato impresso fisicamente con diverse tecniche. Cohen non ha mai definito AARON creativo, ma ha proposto diverse riflessioni a riguardo.

Lavorare con immagini* > *Convoluzione: operazione matematica che può essere applicata anche ad **immagini**. Il **kernel** è una matrice (solitamente piccola, es 3x3) contenente dei valori. La convoluzione viene calcolata facendo "scorrere" lo stesso kernel in tutte le posizioni dell'immagine. Ogni posizione genera **un valore che sarà salvato in un pixel in output.**

Reti neurali: neuroni interconnessi in una rete organizzata a livelli. Cosa succede se volessi un'immagine in input? ***Reti Convoluzionali (CNN)***: reti neurali che effettuano una serie di **convoluzioni in sequenza**. I parametri da ottimizzare sono i **valori contenuti nei diversi kernel** che vengono applicati.



Possono essere applicate anche per creare **modelli discriminativi**. Es. *Task* di classificazione.

AI generativa & Immagini: dal 2014 si sono susseguiti una serie di lavori che propongono di usare diversi modelli di reti neurali per **generare immagini**. Sono stati proposti due modelli fondamentali per lo sviluppo di modelli generativi per dati complessi:

- **Variational autoencoder**
- **Generative Adversarial Network (GAN)**

Sono stati i primi a dare come output non solo la classe di un'immagine, ma **un'immagine vera e propria**.

Generative Adversarial Networks (GAN): due reti neurali ottimizzate assieme:

- **Generator**: generare contenuto realistico (che assomiglia ai dati di training)
- **Discriminator**: decidere se un contenuto è generato o reale (parte del training)

Variational Autoencoder: modello di rete neurale:

- Impara a **riprodurre il suo input** in modo non supervisionato
- Crea un'associazione tra input e **vettore latente, definendo una funzione di probabilità**. Il vettore latente comprime l'input in uno spazio più piccolo seguendo una certa distribuzione. Modificando il vettore latente secondo la distribuzione data, il modello produce in output un contenuto diverso

Stable diffusion: Durante il training viene gradualmente **aggiunto rumore** ai dati. Il processo inverso (denoising) parte da immagini random e **genera nuovi dati con un processo iterativo**.
Text-to-Image models: modelli che usano Stable diffusion possono essere **condizionati da qualche altro input**. Per esempio: del **testo trasformato in latent vector** può essere dato in input al modello per vincolare l'immagine che sarà generata.

LA TRASPARENZA DEGLI ALGORITMI (Lezione 13)
AI e richieste di visto per asilo: Gli algoritmi di intelligenza artificiale filtrano le richieste di visto per il Regno Unito e successivamente i dipendenti utilizzano l'output generato dall'algoritmo per accettare o respingere le richieste. Le domande di visto dei migranti vengono valutate in base alla nazionalità, suddividendo i richiedenti in categorie verde, gialla e rossa. Le persone bianche (categoria verde) hanno una maggiore probabilità di ottenere il visto e il loro processo viene gestito più rapidamente. **Trasparenza e giustificazione**: In che modo l'intelligenza artificiale influenza la decisione finale? Quali sono le motivazioni per l'esito discriminatorio?
Opacità > contesto: Gli algoritmi di machine learning, il cui output è utilizzato per prendere decisioni sulle persone (decisioni con conseguenze rilevanti, ad esempio, richieste di lavoro o di accesso all'università), spesso mancano di trasparenza. **Opacità/mancanza di trasparenza**: Mancanza di conoscenza su come o perché un determinato output algoritmico sia stato prodotto a partire dagli input. **Motivazioni dell'opacità**: (1) È perché l'algoritmo è proprietario? (2) È dovuto alla complessità o al livello tecnico avanzato? (3) Oppure ci sono altre ragioni?
Segretezza: L'opacità può essere una forma di segretezza intenzionale da parte di aziende o governi:

1. Autoprotezione delle aziende per mantenere i segreti commerciali e il vantaggio competitivo.
2. Dibattiti collegati: software proprietario (es. Google, Amazon, Uber, Netflix) contro software open source (es. Unix, Mozilla Firefox).
3. Meccanismi per evitare manipolazioni del sistema, come nel caso dell'ottimizzazione dei motori di ricerca (strategie per ottenere punteggi elevati nei sistemi di classificazione o reputazione).

Protezione delle frontiere: Droni e AI vengono utilizzati dalla Custom and Border Protection (CBP) per controllare i migranti prima ancora che raggiungano i confini degli Stati Uniti. I migranti prendono percorsi più pericolosi per evitare di essere rilevati, aumentando così la probabilità di morte. Raccolta di dati sui migranti senza il loro consenso. Preoccupazioni per la collaborazione con aziende private che sviluppano droni, spesso non controllate da comitati di audit.

Opacità come analfabetismo tecnico: La progettazione di algoritmi e la programmazione sono competenze specialistiche: (1) Introduzione del coding e del pensiero computazionale nell'istruzione obbligatoria; (2) Corsi aperti e pubblici online. (3) Comunicazione di alta qualità per il grande pubblico.
Esperienza e buone pratiche: La programmazione viene considerata un'arte o un mestiere (es. scegliere codice semplice e inserire commenti).

Opacità come discrepanza tra procedure matematiche e interpretazione umana:

1. Non si tratta solo di leggere e comprendere il codice, ma anche di capire come operano gli algoritmi.

2. Complessità inevitabile del modello e dei dati (black box).
3. Le reti neurali sono un esempio tipico di algoritmi a scatola nera.

Reti neurali:
Elementi: (1) Unità interconnesse con input numerici e uno o più output; (2) Regole implicite, estratte dai dati attraverso una serie di equazioni.
Strategia di soluzione: (1) Si ispirano al sistema nervoso, in particolare al modello computazionale di un neurone. (2) Connessioni con diverse importanze. (3) Un neurone somma gli input ricevuti dagli altri neuroni e si attiva se la somma (pesata) supera una certa soglia.

Quando un algoritmo esegue la "programmazione" (calcola in modo ottimale i suoi pesi), ne consegue logicamente che essere comprensibili per gli esseri umani (parte dell'arte di scrivere codice) non è più una preoccupazione, almeno non per il programmatore non umano. Cosa significa che un algoritmo fa la programmazione?

Trasparenza nei sistemi di AI:
Tracciabilità: La tracciabilità di un sistema AI si riferisce alla capacità di tenere traccia dei dati del sistema (raccolta e etichettatura dei dati), dei processi di sviluppo e implementazione, generalmente attraverso documentazione e identificazione registrata.
Spiegabilità: La spiegabilità riguarda la capacità di spiegare sia i processi tecnici di un sistema AI sia le decisioni umane correlate (es. aree di applicazione di un sistema). La spiegabilità tecnica richiede che le decisioni prese da un sistema AI possano essere comprese e tracciate dagli esseri umani. Se il sistema AI ha un impatto significativo sulla vita delle persone, devono essere fornite spiegazioni adeguate delle decisioni basate sull'AI, adattate al livello di competenza dell'utente/soggetto.
Comunicazione: Le persone devono sapere quando stanno interagendo con un sistema AI e dovrebbero avere l'opzione di scegliere un'interazione umana se necessario. La comunicazione riguarda ciò che i sistemi AI possono e non possono fare, incluse la loro accuratezza e i loro limiti, in un modo adatto allo scopo.

ALGORITMI, DISCRIMINAZIONI E SUPERVISIONE UMANA (Lezioni 14 e 15)
Classificazione di genere. In breve: Progetto MIT "Gender Shades" di Joy Buolamwini (2018) Analisi dell'accuratezza intersezionale (genere + razza) di sistemi commerciali di classificazione del genere. Accuratezza elevata complessiva, ma con disparità nei tassi di errore tra i gruppi. Le aziende ottengono i risultati peggiori con donne dalla pelle scura.
Valutazione degli studenti. In breve: Esami del 2020 nel Regno Unito valutati tramite algoritmo di Ofqual. Predizione dei voti basata sul ranking degli studenti dato dai docenti e sui voti storici delle scuole. Alta percentuale di studenti declassati (grandi proteste). Annuncio della cancellazione dell'algoritmo.
Rischio di frode. In breve: Gestione dei benefici sociali (ad esempio, assistenza all'infanzia). Probabilità che un individuo commetta frodi sui benefici. Famiglie penalizzate sulla base di semplici sospetti (spinte verso povertà e depressione). 2021: Multa di 2,7 milioni all'amministrazione fiscale olandese.

Ingiustizia algoritmica **> Definizione di bias algoritmico**: "Sistemi che discriminano in modo sistematico e ingiusto contro certi individui o gruppi a favore di altri, negando opportunità o assegnando esiti indesiderabili su basi irragionevoli." (Friedman e Nissenbaum, 1996).
Due tipi di danni algoritmici
1. **Danni di allocazione**: Decisioni su chi riceve cosa (es. prestiti o offerte di lavoro).
- Problema: Assegnazione ingiusta di risorse/opportunità.
- Quantificabile (metriche di equità).
2. **Danni di rappresentazione**: Rappresentazione del mondo (es. filtri di notizie o raccomandazioni).
- Problema: Rappresentazioni discriminatorie.
- Difficili da formalizzare (problemi di misurazione).

Rappresentazione e Danni di Allocazione
Danni di rappresentazione. Esempio: Ricerca immagini su Google nel 2015 e 2019 per "CEO". Risultati inizialmente discriminatori, successivamente migliorati aggiungendo immagini femminili.
Bias negli embedding delle parole. Esempio: "Uomo è a re come donna è a x". La risposta generata è "regina". Tuttavia, associazioni stereotipate emergono (es. "donna" associata a "casalinga", "uomo" a "programmatore").

Razzismo nascosto nei Modelli Linguistici di Grandi Dimensioni (LLM)
Problema: I modelli linguistici mostrano razzismo implicito, con pregiudizi verso dialetti specifici, come l'African American English (AAE). *Metodologia*: Tecniche di "matched guise" per confrontare l'AAE con l'inglese standard americano (SAE).
Occupazioni e razzismo nascosto: Modelli linguistici collegano certe professioni a dialetti specifici, rivelando bias nei punteggi di occupabilità.

Come sorgono le ingiustizie algoritmiche?
Scenario: Ammissione al college

1. Predizione: Lo studente avrà successo?
2. Decisione: Lo studente sarà ammesso o respinto?

Caratteristiche: Anno di nascita, punteggio esame, razza, ecc. Sensibilità a certe caratteristiche (es. razza) che influenzano i risultati.
Esempio di discriminazione nei punteggi: La classificazione predittiva può portare a errori (falsi positivi o falsi negativi). Gli studenti internazionali tendono a essere respinti ingiustamente più spesso rispetto agli studenti nazionali.
Metriche di equità (esempio). ***Parità statistica***: Probabilità di successo indipendente dalla categoria (es. nazionali/internazionali). Matrice di confusione: Usata per misurare errori e discrepanze tra gruppi.

Trade-off: L'ottimizzazione tra accuratezza ed equità è inevitabile. Esempio: Più errori totali ma minore disparità tra gruppi. La scelta dipende da giudizi etici e politici. La **Pareto Frontier** è il confine formato da tutte le soluzioni che sono Pareto efficienti. Una soluzione è Pareto efficiente se non è possibile migliorare un criterio (es. equità) senza peggiorarne almeno un altro (es. accuratezza). In altre parole, nessuna soluzione sulla frontiera è dominata da un'altra. Esempio pratico: Immagina di voler ottimizzare:

- L'accuratezza di un algoritmo (ad esempio, ridurre gli errori).
- L'equità (ad esempio, ridurre le discriminazioni tra gruppi).

Ciò che accade è che migliorare un aspetto (es. l'accuratezza) potrebbe peggiorare l'altro (es. l'equità). Le soluzioni che trovi lungo la Pareto Frontier sono quelle dove non puoi migliorare uno dei due criteri senza sacrificare qualcosa nell'altro. La Pareto Frontier non dice quale soluzione scegliere, ma mostra tutte le opzioni migliori disponibili, aiutando a confrontare i compromessi tra criteri in competizione.

Obiettivi sociali: I modelli servono per raggiungere scopi più ampi (es. riduzione del tasso di abbandono scolastico). Problemi potenziali:

- Conflitti tra obiettivi (es. massimizzare i voti futuri vs ridurre abbandoni).
- Bias nei dati di addestramento o nei modelli.

Problemi nei dati statistici: ci sono delle scelte che possono avere grandi implicazioni sociali, come quelle dei training data, i modelli e la predizione.

1. Bias di campionamento: Dati non rappresentativi.
2. Bias sociale: Stereotipi incorporati nei dati (es. politiche discriminatorie).

Responsabilità degli attori umani: La responsabilità non può essere delegata solo ai processi tecnici, richiede impegno personale, comprensione delle preoccupazioni altrui e disponibilità a rivedere obiettivi e strategie.
Supervisione umana per garantire che le decisioni guidate dagli algoritmi rispettino principi etici e non causino danni. Mezzi:

1. Meccanismi di governance (es. human-in-the-loop).
2. **Giudizio umano** durante l'intero ciclo di vita dell'AI.
3. **Formazione** e alfabetizzazione sull'AI.

Linee guida per un'AI affidabile:

1. Agenzia umana e supervisione.
2. Robustezza tecnica e sicurezza.
3. Privacy e governance dei dati.
4. Trasparenza (tracciabilità, spiegabilità, comunicazione).
5. Diversità, non discriminazione ed equità.
6. Benessere sociale e ambientale.
7. Responsabilità (auditabilità, riduzione degli impatti negativi).